54

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/713,887	11/16/2000	Paul L. Sinclair	8779	9423

26890 7590 03/13/2003

JAMES M. STOVER
NCR CORPORATION
1700 SOUTH PATTERSON BLVD, WHQ4
DAYTON, OH 45479

EXAMINER	
ALI, MOHAMMAD	
ART UNIT	PAPER NUMBER
2177	

DATE MAILED: 03/13/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/713,887	SINCLAIR ET AL.	
	Examiner	Art Unit	
	Mohammad Ali	2177	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 November 2000* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other:

DETAILED ACTION

1. This communication is responsive to the application filed on November 16, 2000.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1-29 are rejected under 35 U.S.C. 102(e) as being anticipated by Friske et al. ('Friske' hereinafter), US Patent 6,070,170.

As to claim 1, Friske discloses a method for use in managing data in a database system (col. 2, 60-67). Friske teaches 'receiving a request to perform an operation on a set of target data' as the unloaded target data set is reorganized by the processor 106 and loaded into a shadow location 310 of the storage unit 108 (col. 6, lines 25-27 et seq). Further, Friske teaches 'initiating execution of the operation' as a program of machine-readable instructions executable by a digital data processing apparatus to perform a method for reorganizing a database (col. 3, lines 20-22). Finally, Friske teaches 'at some point after execution has begun, placing a lock on the target data to prevent concurrent execution of other operations on the target data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another

process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 2, Friske teaches 'placing an initial lock on the target data at a level that prevents concurrent execution of at least one operation and, at some point after execution has begun, placing a final lock on the target data at a level that prevents concurrent execution of a larger set of operations' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 3, Friske teaches 'the initial lock allows concurrent execution of operations that involve reading the target data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to clam 4, Friske teaches 'the final lock prevents concurrent execution of all operations on the target data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were

completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 5, Friske teaches 'allowing a user to specify the type of lock initially placed on the data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 6, Friske teaches 'the operation is one of the following types: a COLLECT STATISTICS operation, a CREATE INDEX operation, and an ALTER' as after the target data set has been unloaded, the data is ordered in logical sequence in task 410, reorganized in task 412, and loaded into a shadow location in task 414. The target data set may include data indexes which, after the target data set has been reorganized, may be rebuilt into reorganized data indexes in task 416. Rebuilding the data indexes is necessary in the preferred embodiment as discussed above so that quick access to the reorganized data may occur. Log records are applied to the target data set in the shadow location in task 418 which allows any changes to the original

data set which occurred while the reorganization was taking place to be applied to the reorganized target data set (coll. 7, lines 55-67 et seq).

As to claim 7, Friske discloses a database system (col. 2, lines 60-67). Friske teaches 'at least one storage device' as one or more magnetic data storage disks such as a "hard drive" or any other suitable storage device (col. 4, lines 20-23 et seq). Further, Friske teaches 'at least one computing node configured to deliver data to and retrieve data from the storage device' as storage comprises, for example, one or more magnetic data storage disks such as a "hard drive" or any other suitable storage device. The client ('node') computer 102 may include in one embodiment an output module 112 for outputting/displaying program status results on a graphic display 116, print mechanism 114 or data storage medium 118 (col. 4, lines 20-26 et seq). Friske discloses a database-management component (col. 2, lines 60-67). Friske teaches 'receiving a request to perform an operation on a set of target data' as the unloaded target data set is reorganized by the processor 106 and loaded into a shadow location 310 of the storage unit 108 (col. 6, lines 25-27 et seq). Further, Friske teaches 'initiating execution of the operation' as a program of machine-readable instructions executable by a digital data processing apparatus to perform a method for reorganizing a database (col. 3, lines 20-22). Finally, Friske teaches 'at some point after execution has begun, placing a lock on the target data to prevent concurrent execution of other operations on the target data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data

set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 8, Friske teaches 'the database-management system is configured to place an initial lock on the target data at a level that prevents concurrent execution of at least one operation and, at some point after execution has begun, placing a final lock on the target data at a level that prevents concurrent execution of a larger set of operations' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 9, Friske teaches 'the initial lock allows concurrent execution of at least one other operation on the target data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 10, Friske teaches 'the subsequent lock prevents concurrent execution of all other operations on the target data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program

performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 11, Friske teaches 'the database-management system is configured to allow a user to specify the type of lock initially placed on the data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 12, Friske teaches 'multiple computing nodes and multiple storage devices, where each storage node is configured to manage storage of data on at least a subset of the storage devices' as one or more magnetic data storage disks such as a "hard drive" or any other suitable storage device (col. 4, lines 20-23 et seq).

As to claim 13, Friske teaches 'the database-management system is configured to place the lock on a block of data that is spread across more than one of the storage devices' as one or more magnetic data storage disks such as a "hard drive" or any other suitable storage device (col. 4, lines 20-23 et seq).

As to claim 14, Friske teaches 'the operation is one of the following types: a COLLECT STATISTICS operation, a CREATE INDEX operation, and an ALTER TABLE operation' as after the target data set has been unloaded, the data is ordered in logical sequence in task 410, reorganized in task 412, and loaded into a shadow location in task 414. The target data set may include data indexes which, after the target data set has been reorganized, may be rebuilt into reorganized data indexes in task 416. Rebuilding the data indexes is necessary in the preferred embodiment as discussed above so that quick access to the reorganized data may occur. Log records are applied to the target data set in the shadow location in task 418 which allows any changes to the original data set which occurred while the reorganization was taking place to be applied to the reorganized target data set (col. 7, lines 55-67 et seq).

As to claim 15, Friske discloses a computer program, stored on at least one computer-readable storage medium, for use in managing data in a database system, comprising executable instructions that, when executed by a computer(col. 2, 60-67). Friske teaches 'receiving a request to perform an operation on a set of target data' as the unloaded target data set is reorganized by the processor 106 and loaded into a shadow location 310 of the storage unit 108 (col. 6, lines 25-27 et seq). Further, Friske teaches 'initiating execution of the operation' as a program of machine-readable instructions executable by a digital data processing apparatus to perform a method for reorganizing a database (col. 3, lines 20-22). Finally, Friske teaches 'at some point after execution has begun, placing a lock on the target data to prevent concurrent execution of other operations on the target data' as a blocking drain has been used to request a

lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 16, Friske teaches 'the program causes the computer to place an initial lock on the target data at a level that prevents concurrent execution of at least one operation and, at some point after execution has begun, placing a final lock on the target data at a level that prevents concurrent execution of a larger set of operations' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed,

where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13 et seq).

As to claim 17, Friske teaches 'the initial lock allows concurrent execution of at least one other operation on the target data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 18, Friske teaches 'the subsequent lock prevents concurrent execution of all other operations on the target data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task

408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 19, Friske teaches 'the program causes the computer to allow a user to specify the type of lock initially placed on the data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 20, Friske teaches 'the operation is one of the following types: a COLLECT STATISTICS operation, a CREATE INDEX operation, and an ALTER TABLE operation' as after the target data set has been unloaded, the data is ordered in logical sequence in task 410, reorganized in task 412, and loaded into a shadow location in task 414. The target data set may include data indexes which, after the target data set has been reorganized, may be rebuilt into reorganized data indexes in task 416.

Rebuilding the data indexes is necessary in the preferred embodiment as discussed above so that quick access to the reorganized data may occur. Log records are applied to the target data set in the shadow location in task 418 which allows any changes to the original data set which occurred while the reorganization was taking place to be applied to the reorganized target data set (col. 7, lines 55-67 et seq).

As to claim 21, Friske teaches a method for use in managing data in a database system (col. 2, 60-67). Friske teaches 'receiving a request to perform a data-definition operation on a set of target data' as the unloaded target data set is reorganized by the processor 106 and loaded into a shadow location 310 of the storage unit 108 (col. 6, lines 25-27 et seq). Further, Friske teaches 'initiating execution of the operation' as a program of machine-readable instructions executable by a digital data processing apparatus to perform a method for reorganizing a database (col. 3, lines 20-22). Finally, Friske teaches 'at some point after execution has begun, placing a lock on the target data to prevent concurrent execution of other operations on the target data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed,

where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 22, Friske teaches 'the initial lock excludes at least some concurrent operations on the target data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 23, Friske teaches 'allowing a user to select the level of the initial lock' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed,

where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 24, Friske teaches 'placing an initial lock on the target data includes placing one of the following types of locks on the target data an ACCESS lock; a READ lock; and a WRITE lock' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13 et seq).

As to claim 25, Friske teaches 'placing a final lock on the target data includes placing an EXCLUSIVE lock on the target data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task

408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13 et seq).

As to claim 26, Friske teaches 'placing an initial lock on the target data includes locking an entire table' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 27, Friske teaches 'receiving the instruction from the user includes receiving an instruction to perform one of the following operations: a CREATE INDEX operation, a COLLECT STASTICS operation, and an ALTER TABLE operation' as after the target data set has been unloaded, the data is ordered in logical sequence in task 410, reorganized in task 412, and loaded into a shadow location in task 414. The target data set may include data indexes which, after the target data set has been reorganized, may be rebuilt into reorganized data indexes in task 416. Rebuilding the

data indexes is necessary in the preferred embodiment as discussed above so that quick access to the reorganized data may occur. Log records are applied to the target data set in the shadow location in task 418 which allows any changes to the original data set which occurred while the reorganization was taking place to be applied to the reorganized target data set (coll. 7, lines 55-67).

As to claim 28, Friske discloses a method for use in managing data in a database system (col. 2, 60-67). Friske teaches 'receiving a request to perform a MODIFY DATABASE/USER operation on a set of target data' as the unloaded target data set is reorganized by the processor 106 and loaded into a shadow location 310 of the storage unit 108 (col. 6, lines 25-27 et seq). Further, Friske teaches 'initiating execution of the operation' as a program of machine-readable instructions executable by a digital data processing apparatus to perform a method for reorganizing a database (col. 3, lines 20-22). Finally, Friske teaches 'at some point after execution has begun, placing a lock on the target data to prevent concurrent execution of other operations on the target data' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed,

where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13).

As to claim 29, Friske teaches 'maintaining an ACCESS lock on the target database or user and no locks on the immediate parent of the targeted database or user during execution of the MODIFY DATABASE/USER operation' as a blocking drain has been used to request a lock on a target data set for data reorganization purposes. The process requesting the lock, B, would have to wait for the target data set if another process, A, already had a lock on the target data set. If another process, C, came along and requested access to the target data set, it would be placed in a queue behind B waiting for access to the data set. Assuming the reorganization process B and process C were queued and waiting behind A for the target data set, the unload phase shown as task 408 would have to wait to use the target data set until all active logical work units (LUW) of process A were completed, where a LUW includes the processing a program performs between synchronization ('concurrent') points with the apparatus 100 (col. 7, lines 1-13 et seq).

Conclusion

4. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Please see attached PTO-892.

a. US Patent No. 6,460,055 B1, issued to Midgley et al. on 10/01/02. The subject matter disclosed therein is pertinent to that of Claims 1, 7, 15, 21 and 28 (e.g. concurrent operation, target data, execution opeartion etc.).

a. US Patent No. 5,546,579, issued to Josten et al. on 08/13/96. The subject matter disclosed therein is pertinent to that of Claims 1, 7, 15, 21 and 28 (e.g. concurrent operation, target data, execution opeartion etc.).

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mohammad Ali whose telephone number is (703) 605-4356. The examiner can normally be reached on Monday to Thursday from 7:30am-6:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene can be reached on (703) 305-9790. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-9600.

Mohammad Ali

Patent Examiner

March 05, 2003

JEAN P[illegible]MERE
PRIM[illegible]MINER